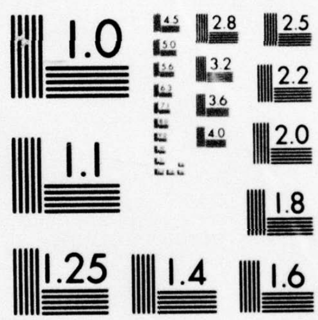
AD-A050 196 PRC INFORMATION SCIENCES CO MCLEAN VA F/G 9/5
RASTER TO LINEAL CONVERSION ANALYSIS.(U)
DEC 77 R L DERRENBACHER, R A NASCI F30602-74-C-0334
UNCLASSIFIED RADC-TR-77-421-VOL-1 NL

| OF |
AD
A050196

END
DATE
FILMED
3-78
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

(2)

AD A050196

RADC-TR-77-421, Volume I (of two)
Final Technical Report
December 1977

RASTER TO LINEAL CONVERSION ANALYSIS

Richard L. Derrenbacher
Richard A. Nasci
Walter H. Winter

PRC Information Sciences Company

Approved for public release; distribution unlimited.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-77-421, Vol I (of two) has been reviewed and is approved for publication.

APPROVED: John R. Baumann

JOHN R. BAUMANN
Project Engineer

APPROVED: H Davis

HOWARD DAVIS
Technical Director
Intelligence & Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

18 19

1. REPORT NUMBER: RADC-TR-77-421, VOL-I
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER:
4. TITLE (and Subtitle): 6 RASTER TO LINEAL CONVERSION ANALYSIS.
5. TYPE OF REPORT & PERIOD COVERED: 9 Final Technical Report, ~~July 1974 - May 1977~~ Jul 74 - May 77,
6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): 10 Richard L. Derrenbacher, Richard A. Nasci, Walter H. Winter
8. CONTRACT OR GRANT NUMBER(s): 15 F30602-74-C-0334 new
9. PERFORMING ORGANIZATION NAME AND ADDRESS: PRC Information Sciences Company, 7600 Old Springhouse Road, McLean VA 22101
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 63701B 17, 16 32020309
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRRP), Griffiss AFB NY 13441
12. REPORT DATE: 11 Dec[redacted]77
13. NUMBER OF PAGES: ~~95~~ 12 138p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same
15. SECURITY CLASS. (of this report): UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: John R. Baumann (IRRP)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Automated Cartography
Raster Processing
Raster to Lineal Conversion
Skeletonization

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report addresses the strong and weak points of past efforts in raster to lineal processing and presents a solution for performing the conversion by reducing the volume and complexity of the data using a skeletonization algorithm. The resultant software system was written in Fortran for running on the Univac 1108.

Volume I is the Final Technical Report.
Volume II is the Program Documentation (Not Published)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

407126

402360

TABLE OF CONTENTS

EVALUATION

The software system developed at RADC under this effort and implemented at the Defense Mapping Agency Aerospace Center (DMAAC) provides a capability to output line center (vector) information from data which has been derived by scanning cartographic color separations on a large format drum raster scanner which is in operation at DMAAC.

This capability will allow DMAAC to significantly reduce the overall time and costs of collecting lineal cartographic feature information. This software system also provides a sound base for further experimentation with more complex raster scanned data.

John R. Baumann

JOHN R. BAUMANN
Project Engineer

I. INTRODUCTION

A. Overview of the Problem

The objective of the Raster-Lineal Conversion Analysis project was to find a method to effectively convert raster scanned cartographic data to a lineal or vector format. The necessity for this stems from the fact that many of the processes involved in manipulating digital cartographic data must have the data in a lineal or cartographic feature oriented form. Gathering the input data is a important problem. Manual lineal digitizers cannot meet the throughput goals of the automated cartographic processes, and their use introduces a certain amount of error in practice because of human operator limitations. Raster scanning devices have the proven ability to accurately extract data from a cartographic manuscript in a matrix or pixel format. The problem that faces us, therefore, is one of cost-effective raster to lineal data conversion.

In order to be cost-effective the raster-lineal conversion process must have a throughput rate significantly higher than that of a lineal digitizing system. The output must be a positionally accurate representation of the scanned graphic input, and it must exhibit an absolute minimum of errors such as feature segmentation, since the time involved in corrective editing has a decidedly detrimental affect on total throughput time.

B. Previously Attempted Solutions

There were two previous major efforts at RADC directed at the raster-lineal conversion problem. The first of these was the Computer Assisted Scanning Techniques (CAST) project. CAST was implemented on the PDP-9 in Macro assembly code. The principle of the linealization technique employed is to establish relationships between disconnected data points within an array formed by the raster points of three consecutive scan lines. Correlations between data points are made within a neighborhood about a given point on a scan line using predetermined criteria for establishing distance and direction between that point and the next. Line segment lists

are constructed, linked, and, when the raster file is exhausted, formatted and output. This system proved unsatisfactory because of deficiencies in the algorithm which resulted in feature segmentation and because hardware limitations allowed for processing only limited amounts of data at one time.

RADC's second major effort in this field, the Computer Assisted Color Separation Project (CACS), is written mainly in FORTRAN for the HIS-635. The underlying principle of the line connection technique is similar to that of CAST with some significant enhancements. CACS is able to contain in core 13 consecutive scan lines simultaneously (as opposed to 3 in CAST). This provides for a much better "look ahead" capability when dealing with wide area data such as those representing horizontal lines and the maxima and minima of curves. Several special-purpose routines within the system are available to recognize and evaluate special problem areas. The use of only 24K of core for the CACS system was made possible by separating the input, line connection, and feature building phase into sequential job activities. In addition, a working feature file, using 16 words of memory for each feature segment processed, was structured to contain all the information required in the feature extraction process to identify a segment and link it to its lineal data and to other segments.

CACS is still somewhat limited in the volume of data that can be converted in one run since it can handle no more than 100 feature crossings per scan line. Also, although the lineal output from CACS in most instances shows an improvement over that of CAST, the error rate is still unacceptably high.

Examination of the input data sets at the points of error revealed that almost all trouble occurred in two areas: first, where the lines lie close to the direction of scan such as on horizontal lines and at the maxima or minima of certain curves, and, second, at some intersections. Next, the algorithms in CACS code were analyzed to determine the cause of the errors. In each case the errors usually could be attributed to a deficiency in one of the special purpose algorithms designed to handle horizontal lines, the maxima or minima of curves, or intersections. Any attempt to fix these problem areas seemed to defy solution since each case was different. Each horizontal line, each maxima or minima, each intersection was

different, if only slightly, from the last. This apparently endless variation in the raster configurations of the problem areas offered little hope of finding an answer that would produce correct results in all cases. It was decided, therefore, that the input data had to be reduced in both volume and complexity before a viable solution could be found, that would yield error free results.

C. Skeletonization

The method chosen by PRC to reduce the raster data is called Skeletonization. This technique is based on what is known in the field of digital image processing as the Golay processes. These Golay processes are generally used to detect the edges of discrete areas in digital images. Our needs, of course, were exactly the opposite; we wished to define a center line in the raster representation of cartographic features rather that the edges. Researchers had already been at work on this problem, experimenting with the Golay processes and similar algorithms to perform specific tasks in their areas of interest.

The algorithm chosen by PRC to solve the problem at hand is based on the work of E.S. Deutsch of the University of Maryland. This algorithm was further developed by PRC to take into account the unique problems associated with manipulating and representing cartographic data. Our algorithm has the following characteristics:

- It "thins" or skeletonizes thick lines into lines one resolution element thick.
- It preserves all continuity information within the original image (all crosses, starbursts, etc., are maintained).
- It removes all single points and a large class of "noise" such as may be introduced during raster scanning.

With this algorithm, raster processing becomes independent of feature segment orientation, and all types of junctions may be readily identified.

The algorithm employs the following notation:

1. Any pixel in the image has eight neighbors N(i) where i=1 through 8. Pictorially they are arranged as follows:

N(4)	N(3)	N(2)
N(5)	Pixel	N(1)
N(6)	N(7)	N(8)

"Odd" neighbors are those along picture axes. "Even" neighbors are along diagonals.

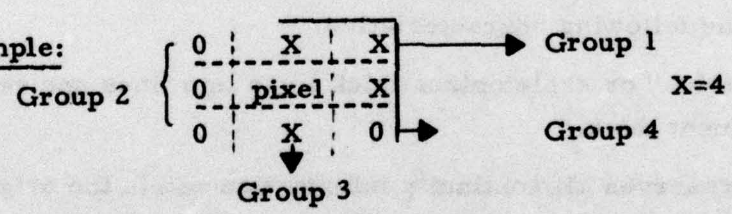
2. Any neighbor N(i) is either darkened (1) or blank (0). The binary value of neighbor N(i) is referenced as $\gamma(i)$ where i=1 through 8 and $\gamma(i)=1$ or 0.

3. The crossing number for any pixel, X, is defined as:

$$X = \sum_{i=1}^{8} \left| \gamma(i+1) - \gamma(i) \right|$$

where $\gamma(9) = \gamma(1)$. Physically, X counts the number of discrete connected groups of neighbors.

Example:



4. The symbol $\wedge$ is the "AND" symbol of binary logic:

$1 \wedge 1=1$

$1 \wedge 0=0 \wedge 1=0 \wedge 0=0$

5. The symbol $\vee$ is the "OR" (inclusive) of binary logic:

$1 \vee 1=1 \vee 0=0 \vee 1=1$

$0 \vee 0=0$

6. The algorithm is parallel across the picture, i.e., pixels removed in pass N are not treated as "removed" until pass N+1.

7. To avoid biasing the skeletonized line, the algorithm alternates between rule A and rule B; odd passes use rule A, even use rule B or vice versa. Only pixels of value "1" need be tested.

RULE A: In order to delete a pixel (change it from "1" to "0") <u>all</u> the following conditions must hold:

If $X=0$ or 2 and $\sum_{i=1}^{8} \gamma(i)=1$

Then $\gamma(1) \wedge \gamma(3) \wedge \gamma(5) =0$

$\gamma(1) \wedge \gamma(3) \wedge \gamma(7) =0$

if $X=4$ and $\sum_{i=1}^{8} \gamma(i) \neq 1$

then $\gamma(1) \wedge \gamma(3) \wedge \gamma(5) =0$

$\gamma(1) \wedge \gamma(3) \wedge \gamma(7) =0$

And

Either $\gamma(1) \wedge \gamma(7) =1$

and

$\gamma(2) \vee \gamma(6) =1$

and

$\gamma(3) = \gamma(4) = \gamma(5) = \gamma(8) =0$

Or $\gamma(1) \wedge \gamma(3) = 1$

and

$\gamma(4) \vee \gamma(8) = 1$

and

$\gamma(7) = \gamma(5) = \gamma(6) = \gamma(7) = 0$

(If $X \neq 0, 2, 4$; no pixel is deleted).

RULE B: In order to delete a pixel all the following conditions must hold:

If $X=0$ or 2 and $\sum_{i=1}^{8} \gamma(i) \neq 1$

Then $\gamma(3) \wedge \gamma(5) \wedge \gamma(7) = 0$

$\gamma(1) \wedge \gamma(5) \wedge \gamma(7) = 0$

If $X=4$ and $\sum_{i=1}^{8} \gamma(i) \neq 1$

Then $\gamma(3) \wedge \gamma(5) \wedge \gamma(7) = 0$

$\gamma(1) \wedge \gamma(5) \wedge \gamma(7) = 0$

and

Either $\gamma(5) \wedge \gamma(3) = 1$

and

$\gamma(6) \vee \gamma(2) = 1$

and

$\gamma(1) = \gamma(4) = \gamma(7) = \gamma(8) = 0$

Or $\gamma(7) \wedge \gamma(5) = 1$

and

$\gamma(8) \vee \gamma(4) = 1$

and

$\gamma(2) = \gamma(6) = \gamma(1) = \gamma(3) = 1$

Although it appears complex, the programming of this part of the algorithm is quite straightforward as each pixel considered requires only the following information:

$\gamma(i) \quad i=1,8$

X

$\sum_{i=1}^{8}\gamma(i)$

All logic is "AND" and "OR".

In actual practice this part of the algorithm was implemented as a table look-up scheme. Since the number of possible configurations for a pixel and its 8 neighbors is only 2^8 or 256, it was possible and practical to set up a table containing a retain or delete decision for each case. Two of these tables are necessary: one for Rule A and one for Rule B. Table entries that indicate retention of a pixel also indicate whether or not it is a junction point and categorize the junction as to type if applicable.

II. SYSTEM CHARACTERISTICS

A. Hardware

The initial Raster-Lineal Conversion System was implemented on a PDP-11/45 in a mixture of Macro Assembler language and FORTRAN. While relatively successful, this system would not support the large-size input format that the RAPS scanner-plotter is capable of producing.

It was then concluded that the system should be implemented entirely in FORTRAN on the Univac 1108 for DMAAC in St. Louis. This was accomplished by first bringing up the system on the Honeywell 6180 at RADC. This was very convenient from the standpoint of accessability and also will provide RADC with a local testbed for future raster-lineal testing and R&D work. The programs thus developed were then transferred (with a minimal amount of necessary changes) to the DMAAC Univac 1108.

The computer resources needed to run a job on the Univac 1108 system are variable within bounds and may be easily changed to meet differing requirements:

- o Core memory -- The amount of Univac 1108 memory required is dependent upon the size and density of the manuscript being processed. It ranges from a minimum of 32K words for an "average" manuscript, such as the Sears Pond contours which are often used for tests at RADC, to a maximum of 60K words for a manuscript with 1500 line crossings per scan line. For very large dense manuscripts even more memory would provide a definite time advantage.

- o Disc Storage -- The amount of disc space needed to run a job is a function of the number of scan lines of data to be processed and the number of features encountered per scan line. It may be calculated by the following formula:

$$D = \frac{Lx2xY}{R}$$

Where: D is the number of words of disc space required,

L is the maximum number of features encountered on any one scan line,

Y is the dimension of the manuscript perpendicular to the scan axis in inches, and

R is the scanning resolution in inches.

This formula provides the space needed to contain the raster data file. Several other small files are also needed, but their sizes are insignificant by comparison.

- Magnetic Tapes -- The system requires two 9-track magnetic tapes for input; another 9-track tape is needed for output (a Xynetics plot tape). It may be mounted on one of the drives previously used for input.

B. Software

The software for the system as configured on the Honeywell 6180 is identical with that on the Univac computer at DMAAC with exception that input is from Automatic Color Separating Device (ACSD) rather than from the RAPS scanner.

The programs for the Raster-Lineal Conversion System are divided into 4 phases or modules each of which performs a specific part of the data conversion process.

- Phase I - Input Module. This group of routines accepts input from two 9-track tapes that are produced by the RAPS raster scanner. The raster data from these tapes is reformatted and output to a random access disc file. Sectioning is performed if required. User supplied parameters such as resolution, scale, etc., are read in and put into a header or communications file along with statistics determined by the Phase I processing.

- Phase II - Skeletonization. The second phase of the system applies the previously discussed skeletonization algorithm to the raster data file repeatedly until all raster images of the cartographic features are reduced to lines of unit thickness.

Since the amount of memory available has practical limits, only a limited piece of the raster file may be in memory at one time. Therefore, a sophisticated "staircasing" function has been developed to control the transfer of data between memory and the disc to minimize the amount of disc I/O necessary. Nevertheless, both the CPU and I/O resource needed by Phase II are a considerable part of the total requirements for the entire Raster-Lineal Conversion System.

During Phase II, all junction points are identified as to type; this information is output onto a disc file.

- Phase III - Linealization. The linealization module first calculates the number of raster scan lines that it can hold in memory at once, taking into account the memory available and the feature density of the manuscript. Strips of this size are read from the raster file and the raster elements are linked by the line following algorithm into segments. Information concerning these segments is saved in a master list until all segments belonging to a cartographic feature have been found, linked and output to disc.

  The amount of bookkeeping necessary to create and maintain these linked feature segment lists is extensive. Thus the greatest amount of code is found in the Phase III Linealization module.

- Phase IV - Output. This final system phase is output. Segment records output in the previous phase are retrieved by following the chains. These segments are then sorted to match up the proper segment end points. When an entire feature has been assembled, it is formatted for the Xynetics Plotter and written to the output tape. During output statistics concerning the features are generated and reported along with a list of all junction points in the file.

A graphic representation of each phase is presented in Figures II-1 through II-4.

A detailed description of each phase in the system and each module within the phase is available in the volume entitled "Raster-Lineal Analysis Project Program Documentation".

A user's guide to assist in the operation of the Raster-Lineal Conversion System is included in this volume as Appendix A.

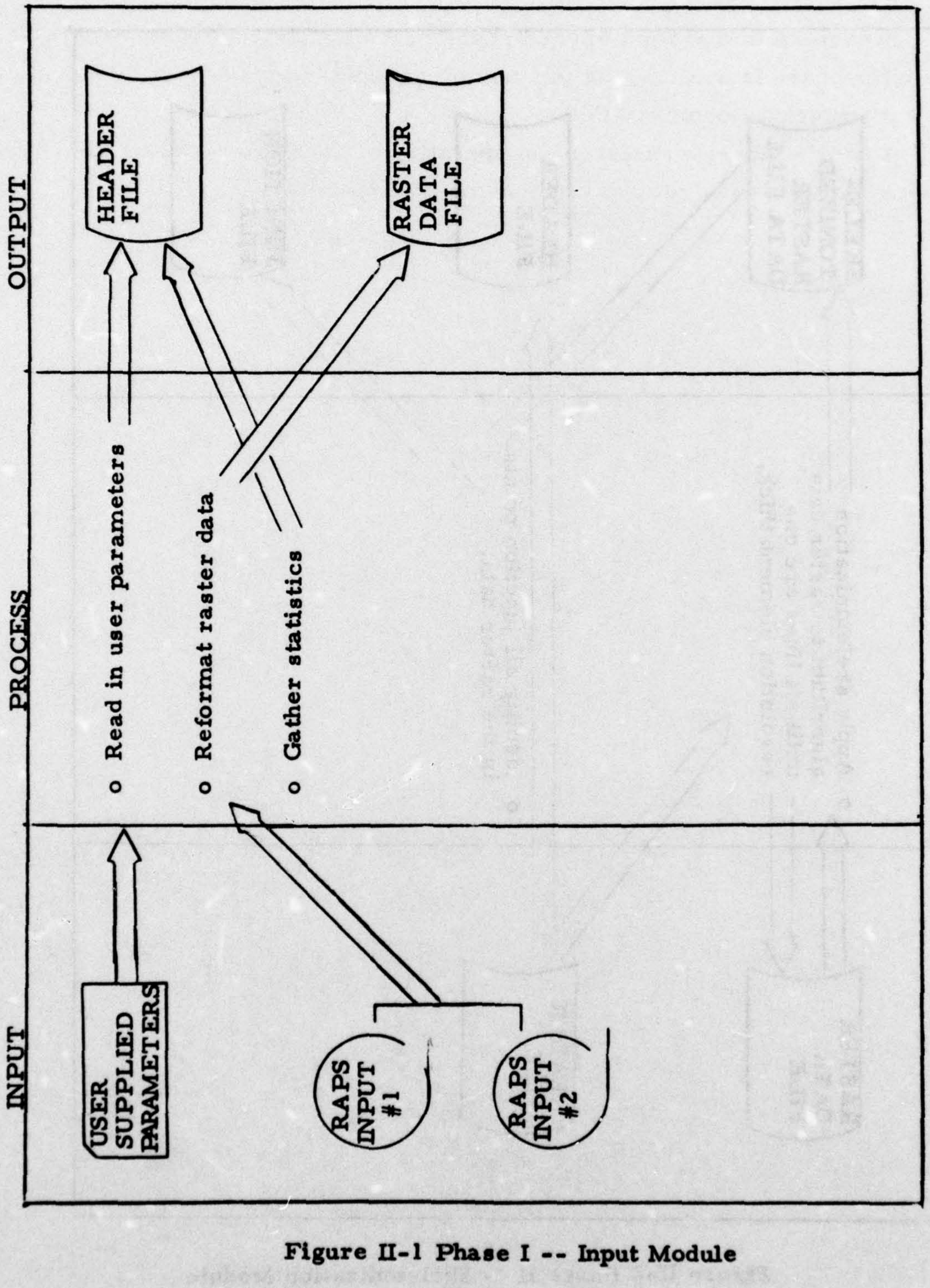


Figure II-1 Phase I -- Input Module

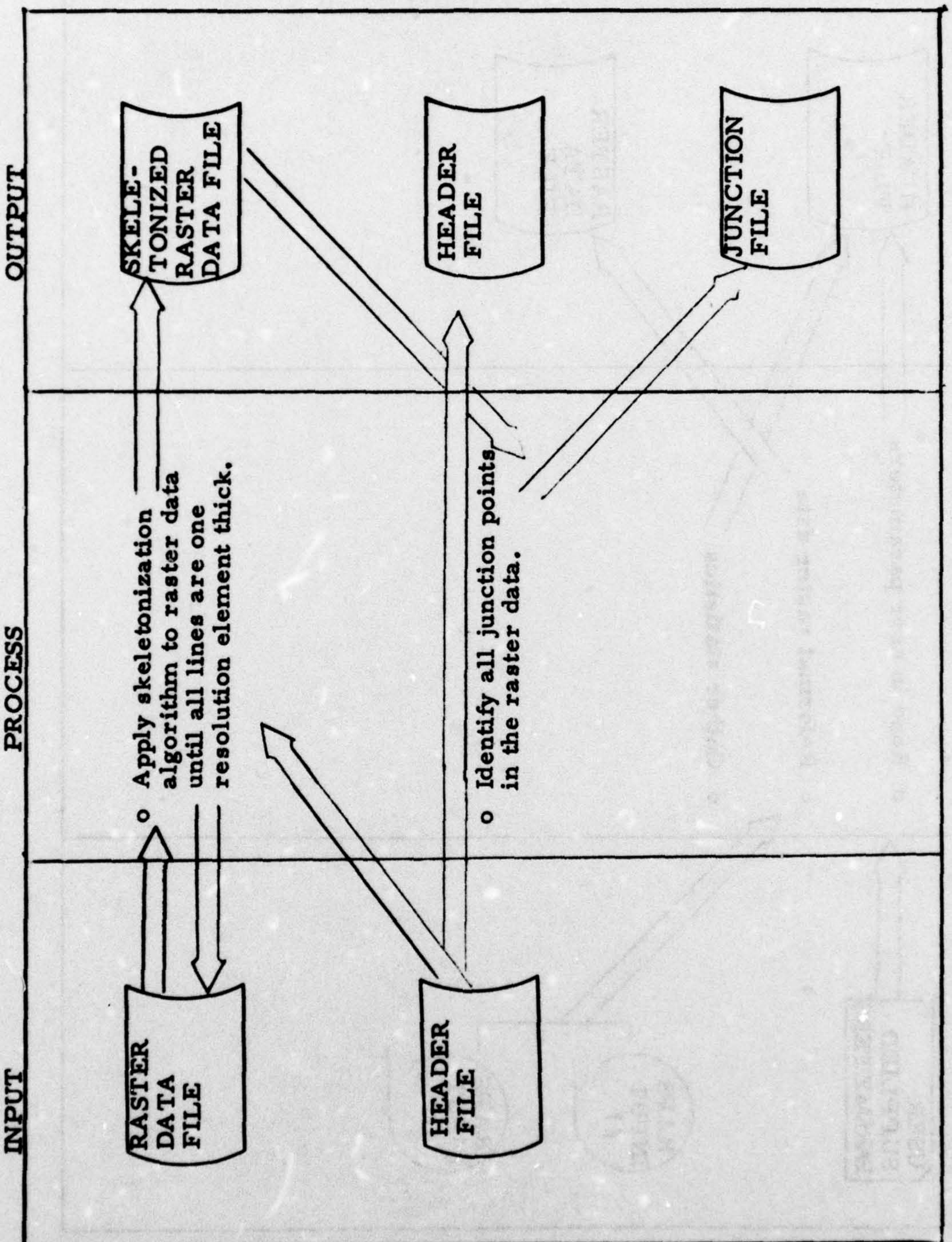


Figure II-2 Phase II -- Skeletonization Module

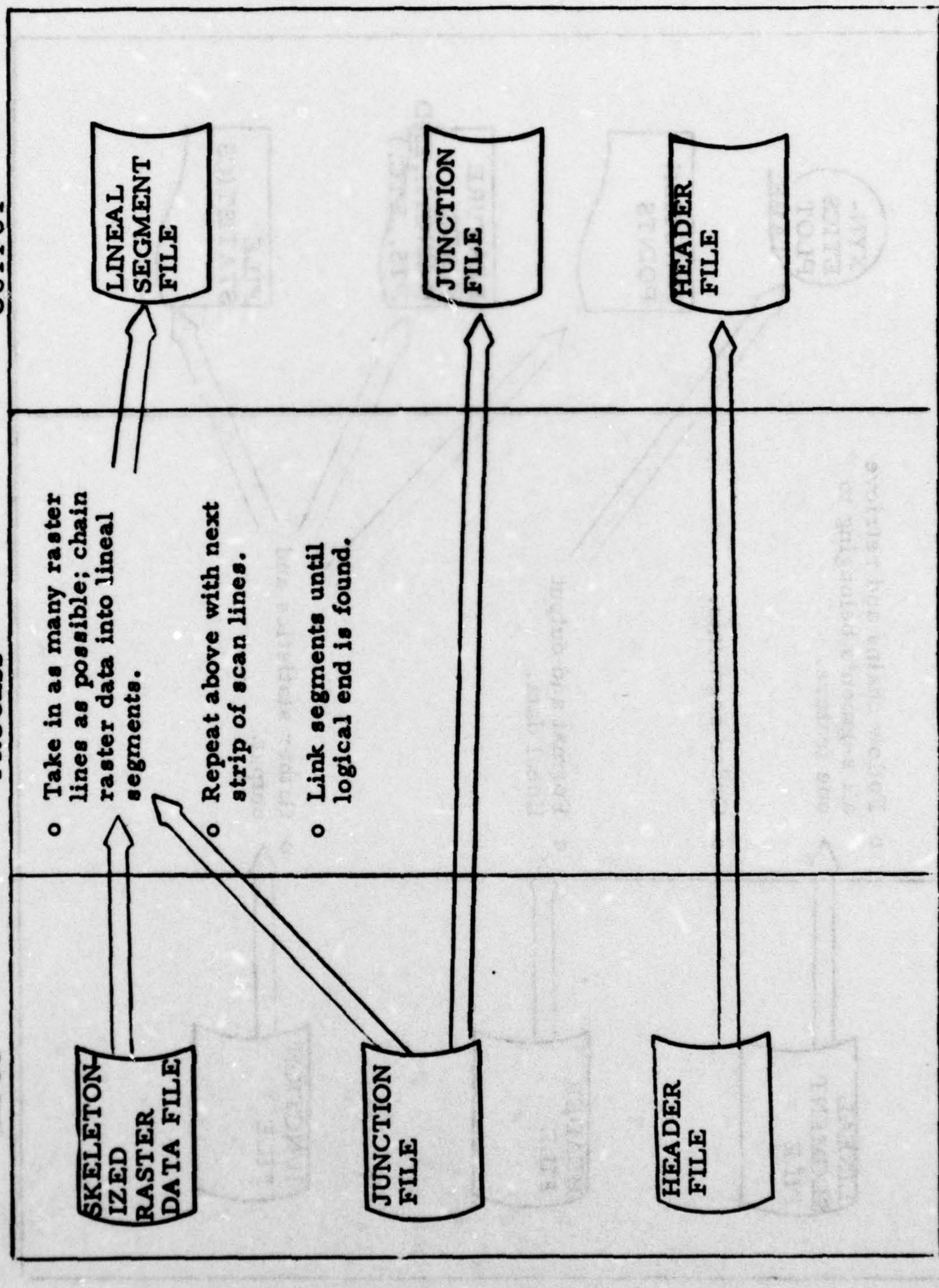


Figure II-3 Phase III-- Linealization Module

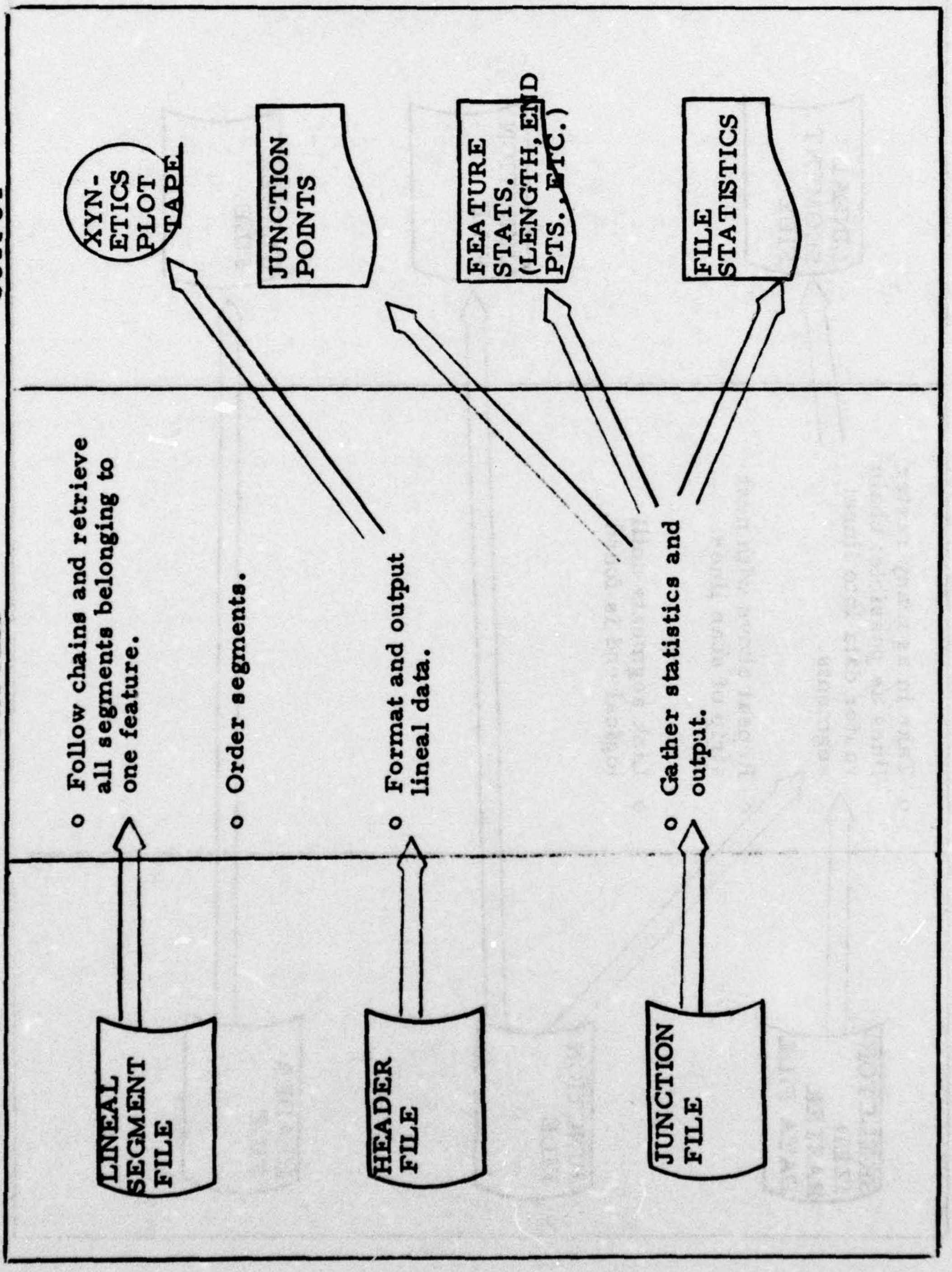


Figure II-4 Phase IV -- Output Module

III. OPERATIONAL CAPABILITIES

A. Input

The Raster-Lineal Conversion System is capable of processing any manuscript that has been scanned with the RAPS scanner. The only restriction is that there not be more than 1500 features (or segments thereof) intersected by any one scan line. This is approximately equal to an average density of 20 features per square inch over 70 inches in the X or scan axis. The local density of features at any one spot on the graphic is not a restriction as long as they have been properly separated by the scanner. In the unlikely event that a manuscript must be processed which exceeds the above restriction, the system is capable of sectioning the manuscript so that the job can be run in 2 or more parts.

B. Throughput

A lack of both time and good test data has thus far prevented extensive testing of the system implemented on the Univac 1108. The RAPS scanner was not operational until very late in the contract. The data sets that we were able to pass through the system were too small to provide any meaningful results.

Some timings, however, are available from the testbed Honeywell system. An experiment was conducted on the hand-drawn Sears Pond contour sheet which is often used for testing at RADC. This sheet, which is about 18" x 22" in size and contains approximately 1650" of lineal feature data, was scanned at a 4 mil resolution. This manuscript was processed by the system using 44 minutes of CPU and 30 minutes of I/O time. This yields a processing rate of around 37" of lineal data per minute of CPU time on the Honeywell 6180 for this data set.

These numbers may not be entirely accurate taken out of context. It must be remembered that several factors affect throughput rates for the system:

- o Feature density -- Given graphics of the same size with features of the same line weights, more dense ones are processed more efficiently than less dense ones. This is due to the system overhead involved in moving scan line data between disc files and memory. Since the system operates on groups of scan lines, the more lineal inches of data in each block, the more efficient the operation, especially with regard to I/O time.
- o Line weight -- The thickness of lines representing features affects only one part of the system, Phase II. This effect, however, is dramatic, and Phase II typically accounts for 2/3 of the total system processing time. A line of 4-mil thickness will take only half as long to reduce to unit thickness as one 8 mils thick. It is also true that the entire raster file is repeatedly passed until all raster data is one resolution unit thick. Therefore, even a few thicker lines, blobs, heavy lettering, etc., will have a very adverse effect on total throughput.
- o Graphic size and resolution -- These two factors translate into one thing: the total number of scan lines of data the system must process. As already mentioned there is considerable system overhead incurred in processing each line of raster data. Thus scanning a rectangular manuscript with the short dimension parallel to the scan axis, or scanning at a higher resolution than necessary to accurately capture the data, will be detrimental to total system throughput.

C. <u>Limitations</u>

At the end of the contract one known bug remained in the system. The problem, which we tried but were unable to pinpoint, is located somewhere in the segment processing code. This anomaly manifests itself by dropping one segment from a few features. It occurs very infrequently.

Because random file definition takes place at execution time, the record size must be fixed when the programs are compiled. The sizes of system arrays used to buffer these records are also fixed. During testing of the

system, these parameters were set to accommodate up to 250 line crossings rather than 1500. This uses less system resources and results in better job turnaround. These parameters may be changed to anything up to the 1500 limit by following the directions in Appendix B. It is suggested, however, that they be left where they are until the system has been thoroughly tested on the more modest data sets.

IV. RECOMMENDATIONS

PRC's recommendations concerning the Raster-Lineal Conversion System fall into two areas: testing and improvement.

A. System Tests

It is recommended that a set of test data, of known quality and quantity, be generated and used to exercise the system. We suggest that those methods outlined in the Test Plans & Procedures document could serve as the basis for several carefully controlled experiments.

The purpose of testing the Raster-Lineal Conversion System is two-fold. First, of course, it is desirable to measure the throughput of the system in terms of how much data can be processed relative to the amount of computer resources needed to do the job. The second reason is to analyze the performance of the pieces of the system's software relative to the whole. In other words, what relationship do the workings of each separate phase of the system have to the overall process. This information is vital to an understanding of the problem of improving this and future systems of similar purpose.

B. System Improvements

Several areas of system improvement and enhancement deserve attention.

- Fix error -- The one known system program bug should be fixed.
- Optimization -- The code comprising the Raster-Lineal Conversion System programs should be studied for the expressed purpose of optimizing it to improve throughput speed. Special attention should be given to Phase II methodology to attempt to eliminate unnecessarily processing lines of raster data on which no skeletonization remains to be done. Also, consideration should be given to the possibility of eliminating junction points from the raster file as they are recognized. This would make

Phase III processing somewhat simpler if it never had to look for them. The elimination of these points should have little affect on the output since features are always broken there anyway.

- Auto edit -- An auto edit capability should be considered for implementation. This would be used to join breaks in contours and to eliminate various types of noise that appear as very short features. Implementation of this system enhancement should not require an extensive programming effort since all the needed data (end point and length statistics) is currently being collected.

APPENDIX A
USER'S GUIDE

This appendix is a guide to potential users of the Raster-Lineal Conversion System. Familiarity with the Univac 1108 control card structure is assumed.

A. Input

Two raster scan tapes from the RAPS are needed for input to the Raster-Lineal Conversion System. One contains the even numbered scan lines, and the other contains the odd numbered scan lines.

The user is also required to specify, via data cards, the input format, output format, scanning resolution, plot scale, sectioning parameters, and and maximum number of scan lines on the input tapes. These are clarified below.

B. Job Execution

Figure A-1 represents a deck of control cards that will execute the system. Each control card has been numbered, and the following explanations are keyed to those numbers.

```
@RUN, /R
@HDG UNCLASSIFIED
@ASG,A DBM*UE20-DAVIS.
@USE T.,DBM*UE20-DAVIS.
@COPY,SR T.,TPF$.
@PACK T.
@PREP T.
@ASG,T 21,U9H,AAAAA
@ASG,T 22,U9H,BBBBB
@ASG,T 2,F/25/POS/25
@ASG,T 3,F/1/TRK/1
@ASG,T 4,F/1/POS/2
@MAP,IL
@IN ANALY
@XQT
  $USER
  INMAT=1,
  OUTTYP=1,
  RES=.00098425,
  SCALE=1.0,
  SECT=1,
  SMINX=1,
  SMAXX=20000,
  SMINY=0,
  SMAXY=5000,
  MAXREC=5001
  $END
@PMD,EL
@FREE 21
@FREE 22
@MAP,IL
  IN SKELET
@XQT
@PMD,EL
@MAP,IL
  IN EXEC3
@XQT
@PMD,EL
@ASG,T 23,U9H,CCCCC
@MAP,IL
  IN PHAS4
  LIB DBM*XYNETICSPLOT.
@XQT
@PMD,EL
@FIN
```

Figure A-1 Control Deck

- Cards 1-7 -- These are the standard beginning of job cards for this type of run. DBM*UE20-DAVIS is the name of the catalogued disc file where both the source and object code for the Raster-Lineal Conversion System programs are located.

- Card 8 -- This card is used to assign the RAPS scanner tape (AAAAA) with the even scan line numbers to file 21.

- Card 9 -- This card is used to assign the tape (BBBBB) with the odd numbered lines to file 22.

- Card 10 -- File 2 is used for the random Raster Data file. The number of disc positions necessary to contain this file may be calculated by the formula given in Section II-A above. This same file space is overwritten by the Lineal Segment file in Phase III, but this will always be smaller than the Raster Data file.

- Card 11 -- File 3 is the Header file used for intermodule communication. It is fixed length, 20 words, and thus 1 track is more than enough room.

- Card 12 -- File 4 is the junction file. Two Positions should be enough for any manuscript.

- Cards 13-28 -- These cards caused the Phase I routines to be mapped and executed. Cards 16-27 are data cards used by a FORTRAN Namelist read. Specifically:

  - Card 16 -- The name of the namelist is USER.
  - Card 17 -- INMAT is the variable used to contain the input format type. INMAT should equal 1 for the RAPS scanner input.
  - Card 18 -- OUTTYP=1 designates that output is to a Xynetics plotter tape.
  - Card 19 -- This gives the scanning resolution in inches. Thus RES=.00098425 means the resolution is 25 microns.
  - Card 20 -- This card is used to set the scale at which the output is to be plotted. This is usually left at 1.0 since scale may also be adjusted when actually plotting.

- Card 21 -- SECT=0 means that no sectioning is wanted and that the entire manuscript is to be processed. SECT=1 signifies that the data is to be sectioned.
- Cards 22-25 -- These cards contain the sectioning parameters in resolution units. SMINX is the minimum X-coordinate and must be 1 or larger. SMAXX is the maximum X-coordinate wanted. SMINY is the first scan line to be considered for processing. SMAXY is the last scan line number to be processed. These four cards are only valid if SECT=1.
- Card 26 -- MAXREC sets the number of records in the Raster Data file to be allocated in Phase I. One record is needed for each scan line to be processed.
- Card 27 -- End of Namelist input.

o Cards 29, 30 -- These control cards free up the input tape drives after Phase I is completed.

o Card 31-34 -- These 4 cards cause Phase II to be mapped and executed.

o Cards 35-38 -- These 4 cards cause Phase III to be mapped and executed.

o Card 39 -- This card is used to assign the Xynetics plot tape (CCCCC) to the output file, 23.

o Cards 40-44 -- These 5 cards cause Phase IV to be mapped and executed. DMB*XYNETICSPLOT is the name of the catalogued file containing the Xynetics library plot routines.

o Card 45 -- End of job stream.

C. Output

There are two principal outputs from the Raster-Lineal Conversion System. The first of these is the Xynetics plot tape which contains the linealized features. Edit circles are also contained on the tape to indicate visually the places where junctions were encountered. This tape is to be plotted in the usual manner on the Xynetics plotter. When the lineal features have finished plotting, the plotter system will pause and the message "EDIT CIRCLES" will appear on the system teletype. If the user

wishes to have these edit circles plotted, he must now command the system to continue. If he does, the plotter will switch pens and draw $\frac{1}{4}$" circles around each junction point.

The second system output is the statistical lists. For each individual feature the following information is given: 1) position within the output file, 2) feature length in inches, 3) number of coordinates in feature, 4) feature end point coordinates. Each junction point is listed with its position coordinates.

Finally, statistics for the entire file with regard to number of features, total length of the lineal features and total number of junctions found are printed.

APPENDIX B
BUFFER SIZE ALTERATION

The alteration of buffer array sizes in the Raster-Lineal Conversion System requires some understanding as to their individual uses.

First of all, a breakdown of the system by phases must be shown.

A. Phase I

This is the input phase of the system. It is designed to read scanner data and convert it to start/stop pairs. Functionally, it serves as a statistical package for use by subsequent phases of the system. Also, it sets up a disk file to be used by the Phase II Skeletonization process. Another task performed by Phase I is the sectioning of data in any direction. Once sectioned, only the desired data is written to the disk for subsequent processing.

Since this is the initial stage of the system's processes, there are no statistics available on the incoming data. Because the random record disk file is defined at execution time, there is a need for a compilation time parameter. This parameter is RECSIZ. Its value is twice the maximum number of line crossings on any line of the input data. Since this figure cannot be determined until the final line is input, this parameter must be "hard-wired" into the software.

Our recommendation is to leave this parameter at a size which will accommodate the largest line crossing data file which will be put through the system.

B. Phase II

Phase II of the Raster-Lineal Conversion System performs line thinning to reduce the data to be handled by subsequent phases. The original data file set up in *Phase I is* skeletonized to unit thickness. The processed file contains line center data. Also, skeletonization creates a junction file for use in the Phase III Linealization process.

At this point in the system's process, Phase I has yielded some statistics on the original data file. However, once again, there is a parameter which must be "hard-wired". This parameter, RECSIZ, defines the buffer size for inputting the already defined random disk file. Since this data is double-buffered, the core overhead is minimal. The minimization of core overhead can only be seen if the main program, SKELET, of Phase II gets recompiled before execution.

From Phase I, the maximum number of line crossings on any line of the input scanner data file may be found in the header file. From this number, the dimension of the secondary (working) input file may be altered. The recompilation of SKELET prior to loading and execution of the system will afford the minimal core requirement for this phase of the system's processes.

Another unknown arises in Phase II. This is the maximum number of junctions found on a line. This quantity cannot be known until skeletonization has completed its task. Thus, another parameter must be "hard-wired". This parameter, TOP, defines the dimension of a buffer for junction file creation.

Phase II will save the statistics on the junction file for subsequent processing by Phase III Linealization.

C. Phase III

Phase III Linealization processes the line center data file set up by the Phase II Skeletonization module. Its task is to follow the lines, forming lineal features to be plotted via the Phase IV Output routine.

To allow flexibility in core allocation, Phase III Linealization defines one large array to be shared by five different files throughout its execution. From statistics accumulated to this point, the main program for Phase III, EXEC3, can allocate the array in a very efficient manner. After allocating three arrays out of the master array, EXEC3 will break up the remainder dynamically. The remainder is made up of the raster file (skeletonized) buffer and the junction file buffer.

Again, the maximum number of line crossings in the raster file must be set in EXEC3 prior to execution. Also, the main array size may be varied to reduce overhead on smaller raster files. Again, a program needs to be recompiled prior to execution in order to afford efficient system performance.

APPENDIX C

HONEYWELL 6180 VERSION OF THE SYSTEM

This appendix is an aid to users who may wish to use the Raster-Lineal Conversion System as it exists on the Honeywell 6180 at RADC. This version runs under the GCOS operating system. All source code, object code, and execution files are maintained under GCOS on catalog BICDZC05, which can be accessed either by over-the-counter card input or from a remote terminal.

A. Input and Output

An output tape from the ACSD is required for input to this version of the Raster-Lineal Conversion System. Otherwise, the inputs and outputs are exactly the same as described in sections A and C of Appendix A for the system on the Univac 1108.

B. Job Execution

Figure C-1 shows a deck of control cards that will cause the system to execute. A file containing this deck is available on GCOS under catalog BICDZC05. The file name is E.RTOL.

For the purpose of illustration, the control cards are divided here into four parts. In the first group are those necessary to execute Phase I; the second group is for Phase II, etc. This grouping is convenient for purposes of testing and experimenting with the system. By saving certain intermediate files, the later phases can be run and tested, altered, and then rerun without the expense of executing earlier phases.

Each group of control cards is explained below as necessary. A familiarity with the GCOS control card structure is assumed.

- Cards 1-3 -- These are used to initiate job execution.

- Cards 4-10 -- These execute a COBOL program, ACSDIN, which was developed under another effort to read and reformat ACSD tapes. Cards 8 and 9 are for identifying the 24-bit ACSD tape and Card 10 defines the output disc file.

- Cards 11-20 -- These cards execute the rest of Phase I. Card 18 causes the output file of the previous program, ACSDIN, to be released. This file is used as input to this part of Phase I. Card 19 defines the format raster output of Phase I. Card 20 defines the header file. Both of these files have been placed on 180 disc packs for convenience in experimentation.

- Cards 21-32 -- These are FORTRAN Namelist input cards. Their use is the same as those explained under the UNIVAC 1108 system description except that Card 22 (INMAT) must be equal to 2 for ACSD input and Card 31 (DEBUG) is used only for system debugging purposes and should normally be set to 0.

- Cards 33-40 -- These control cards execute Phase II of the Raster-Lineal Conversion System. The skeletonized raster data is placed on file 02 (Card 38). Card 40 defines the junction file which is used in later phases.

- Cards 41-70 -- Phase III, linealization, is executed by these cards. File 01 (Card 67) is used to contain the lineal segment file. The skeletonized raster data is no longer needed and the file is released (Card 68).

- Cards 71-88 -- These cards execute the final phase of the system and wrap up the job. Cards 74, 75, and 76 are used to load the Xynetics plot routines into the execution module. Cards 83 and 84 define the Xynetics plotter output tape to the system.

```
$     IDENT      BICDZC05,PRC-DERR , 320203090334
$     USERID     BICDZC05$XXXXXX
$     MSG1       C,REQUEST 1 HOUR OF CPU TIME.
$     OPTION     COBOL
$     SELECT     BICDZC05/0.ACSDIN
$     EXECUTE
$     LIMITS     10,10K,-3K
$     TAPE       A1,X11D,S0475,,ACSDIN,,DEN5
$     FFILE      A1,NSTDLB,NBUFF/2,BUFSIZ/1024,FXLNG/1024,NOSRLS
$     DISC       B1,B1S,10
$     OPTION     FORTRAN
$     SELECT     BICDZC05/0.EXEC1
$     SELECT     BICDZC05/0.INPUT1
$     SELECT     BICDZC05/0.ANALY
$     SELECT     BICDZC05/0.SECTON
$     SELECT     BICDZC05/0.OUTPT1
$     EXECUTE
$     DISC       01,BIR,10
$     180PK      02,B2S,R,18005,,,1/400
$     180PK      03,B3S,S,18005,,,1001/5
$     USER
      INMAT=2,
      OUTTYP=1,
      RES=2.0,
      SCALE=1.0,
      SECT=0,
      SMINX=0,
      SMAXX=10000,
      SMINY=1000,
      SMAXY=4000,
      DEBUG=1,
$     END OF USER INPUT DATA
```

Figure C-1 Phase I Control Cards (Page 1 of 4)

```
33   $   OPTION    FORTRAN
34   $   SELECT    BICDZC05/0.SKELET
35   $   SELECT    BICDZC05/0.KELIN
36   $   EXECUTE
37   $   LIMITS    100,40K
38   $   180PK     02,B2S,R,18005,,,1/400
39   $   180PK     03,B3S,S,18005,,,1001/5
40   $   180PK     04,B4S,S,18005,,,401/99
```

Figure C-1 Phase II Control Cards (Page 2 of 4)

```
$      OPTION     FORTRAN
$      LOWLOAD
$      SELECT     BICDZC05/0.XEC3
$      SELECT     BICDZC05/0.LINEA
$      SELECT     BICDZC05/0.SEGBL
$      SELECT     BICDZC05/0.JUNC
$      SELECT     BICDZC05/0.JPRO
$      SELECT     BICDZC05/0.LINE1
$      SELECT     BICDZC05/0.LINEN
$      SELECT     BICDZC05/0.MIDLI
$      SELECT     BICDZC05/0.CLSFE
$      SELECT     BICDZC05/0.SCAN
$      SELECT     BICDZC05/0.FLAG
$      SELECT     BICDZC05/0.VECPA
$      SELECT     BICDZC05/0.RUN
$      SELECT     BICDZC05/0.CHAIN
$      SELECT     BICDZC05/0.SGMFIL
$      SELECT     BICDZC05/0.FLUSH
$      SELECT     BICDZC05/0.INSRTX
$      SELECT     BICDZC05/0.SERCHX
$      SELECT     BICDZC05/0.DATFIL
$      SELECT     BICDZC05/0.VECFIL
$      SELECT     BICDZC05/0.RECOUT
$      SELECT     BICDZC05/0.MTRFIL
$      EXECUTE
$      LIMITS     30,40K,,10000
$      DISC       01,X1S,10R
$      180PK      02,B2R,R,18005,,,1/400
$      180PK      03,B3S,S,18005,,,1001/5
$      180PK      04,B4S,S,18005,,,401/99
```

Figure C-1 Phase III Control Cards (Page 3 of 4)

```
71   $   LOWLOAD
72   $   OPTION    FORTRAN
73   $   SELECT    BICDZC05/0.PHAS4
74   $   SELECT    BICDZC05/0.XYN
75   $   SELECT    BICDZC05/0.MOVECH
76   $   SELECT    BICDZC05/0.MOVEA9
77   $   SELECT    BICDZC05/0.OUTPUT
78   $   SELECT    BICDZC05/0.POINTS
79   $   SELECT    BICDZC05/0.CREATE
80   $   EXECUTE
81   $   LIMITS    30,50K
82   $   FFILE     05,NOSRLS,FIXLNG/128,BUFSIZ/129,ASA9
83   $   TAPE9     05,X1D,S0423,,XYNOUT,,DEN8
84   $   DISC      01,X1R,10R
85   $   180PK     03,B3R,S,18005,,,1001/5
86   $   180PK     04,B4R,S,18005,,,401,99
87   $   ENDJOB
88***EOF
```

Figure C-1 Phase IV Control Cards (Page 4 of 4)